

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ  
Декан ФИиВТ

УТВЕРЖДАЮ /А.А. Кречетов/  
(Ф.И.О. декана (директора института))

15.01.2025 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Б.1.1.16 Объектно-ориентированное программирование

(код и наименование дисциплины по учебному плану)

Направление подготовки  
(специальность)

09.03.01 Информатика и вычислительная техника

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Компьютерный дизайн

Курс

2

Семестр

3, 4

**Распределение учебного времени**

Трудоемкость по учебному плану	324 / 9	часов/зачетных единиц
Лекции	18	часов
Лабораторные работы	126	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	144	часов
Контактная работа по экзамену	6	часов
Курсовой проект (работа)	4	семестр
Самостоятельная работа обучающихся (без учета экз.)	144	часов
Самостоятельная работа по подготовке к экзамену	30	часов
Экзамен	4	семестр
Зачет	3	семестр
БРК, ДЗ	-	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.01 Информатика и вычислительная техника

Программу составили:

доцент с ученой степенью кандидата наук	ИВС	СОГЛАСОВАНО	В.Б. Малашкевич
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина  
Кафедра информационно-вычислительных систем

(наименование кафедры)			
14.01.2025	протокол №	19	
(дата)			
Заведующий кафедрой	СОГЛАСОВАНО	Д.В. Морохин	
		(И.О. Фамилия)	

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)  
кафедрой(ами).  
СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Д.В. Морохин
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит  
выпускающая кафедра

СОГЛАСОВАНО	А.А. Кречетов
	(И.О. Фамилия)

Эксперт(ы): Усков Юрий Викторович, Генеральный директор ООО «Ричмедиа»

Рабочая программа проверена и зарегистрирована в УМЦ 04.02.2025 г.  
Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

## Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	<b>знания:</b> Знать алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения <b>умения:</b> <b>навыки:</b>
	ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	<b>знания:</b> <b>умения:</b> Уметь составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули <b>навыки:</b>
	ОПК-8.3. Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	<b>знания:</b> <b>умения:</b> <b>навыки:</b> Владеть языком программирования; навыками отладки и тестирования работоспособности программы

## Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Основы программирования (ОПК-8)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих практиках: Преддипломная практика (ОПК-8); государственной итоговой аттестации в форме: Подготовка к сдаче и сдача государственного экзамена (ОПК-8)

## Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические и лабораторные занятия

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, классическая лекция, мини-проекты, проблемная лекция

#### Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 3 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
<b>Основы объектно-ориентированного подхода к программированию</b>	<b>144</b>	<b>ОПК-8</b>
Лекция. Применение ООП в проектировании информационных систем	1	
Лекция. Объектно-ориентированный подход к проектированию ПО. Основные принципы ООП	1	
Лекция. Язык программирования C++. Особенности, Отличия от языка C	1	
Лекция. Управление памятью в C++. операторы new и delete	1	
Лекция. Ввод-вывод в C++. Потоки istream и ostream. Операторы<< >>	1	
Лекция. Инкапсуляция. Понятие объекта. Классы	1	
Лекция. Функции-члены класса. Способы описания. Оператор принадлежности ::	1	
Лекция. Конструкторы и деструкторы	1	
Лекция. Конструктор по-умолчанию, конструктор копирования	1	
Лекция. Определение полиморфизма. Перегрузка функций.	1	
Лекция. Перегрузка операций.Шаблоны	1	
Лекция. "Дружественные" функции и "дружественные" классы	1	
Лекция. Наследование. Базовые классы. Производные классы	1	
Лекция. Виртуальные функции. Абстрактные классы.Иерархии классов	1	
Лекция. Конструкторы производных классов	1	
Лекция. Стандартная библиотека.Строки.Итераторы.Контейнерные классы	1	
Лекция. Объектно-ориентированные среды разработки ПО. RAD.	2	
Лабораторная работа. Методы сборки программ на языке C++	4	
Лабораторная работа. Классы. Методы класса. Уровни инкапсуляции..	4	
Лабораторная работа. Конструкторы и деструкторы	4	
Лабораторная работа. Конструктор по-умолчанию. Конструктор копирования	4	
Лабораторная работа. Перегрузка методов	4	
Лабораторная работа. Класс с перегруженными конструкторами и методами	4	
Лабораторная работа. Перегрузка операций	4	
Лабораторная работа. "Дружественные" функции. Перегрузка	4	

операций ввода-вывода.		
Лабораторная работа. Наследование. Базовые и производные классы.	4	
Лабораторная работа. Использование конструкторов базового класса в производном классе	4	
Лабораторная работа. Строковые классы.	4	
Лабораторная работа. Контейнерные классы и итераторы	4	
Лабораторная работа. Построение простого GUI-приложения с помощью OO RAD	6	
Задания для самостоятельной работы, в том числе выполнение КР, РГР		
Разработать клиентское WEB-приложение с использованием объектно-ориентированных средств и языков.	72	
Иная контактная работа: зачет, выполнение контрольной работы	0	

#### 4 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
<b>ООП ч.2</b>	<b>108</b>	<b>ОПК-8</b>
Лабораторная работа. Среда разработки Microsoft Visual Studio Общие принципы разработки. Разработка консольных приложений. Разработка статических классов. Обработка ошибок.	4	
Лабораторная работа. Разработка простейших программ с графическим интерфейсом с применением статических классов.	4	
Лабораторная работа. Инкапсуляция и наследование. Разработка классов и простых приложений с их использованием.	8	
Лабораторная работа. Полиморфизм. Перегрузка методов. Виртуальные методы. Абстрактные классы.	8	
Лабораторная работа. Полиморфизм. Переопределение методов. Раннее и позднее связывание.	8	
Лабораторная работа. Графические форматы. Доступ к пикселям изображения. Разработка программ для работы с графикой.	8	
Лабораторная работа. Виды меню в приложении. Разработка каркаса приложения.	8	
Лабораторная работа. Классы и перегрузка операций.	8	
Лабораторная работа. Принципы событийно-ориентированного программирования. Разработка приложений с обработкой событий.	8	
Лабораторная работа. Многопоточные приложения.	8	
Задания для самостоятельной работы, в том числе выполнение курсового проекта/работы Разработка методов класса приемами структурного программирования.		
Написание OO GUI-приложений с использованием MS Visual Studio	36	
выполнение курсового проекта/работы	36	
Иная контактная работа: защита курсового проекта/работы	0	

Подготовка к экзамену	30
Проведение экзамена	6

## Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности. **Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **занятиям семинарского типа** включает ознакомление с планом лабораторного занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины.

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Изучение дисциплины включает выполнение контрольной работы, лабораторных работ, а также РГР и курсовой работы. Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе. Формой промежуточной аттестации по дисциплине является экзамен и зачет.

Формой промежуточной аттестации по курсовой работе является

## Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
<b>УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ</b>		
1.	Давыдов, Владимир Григорьевич. Технологии программирования С++ [Текст] : [учеб. пособие для студентов вузов по специальности 210100 "Упр. и информатика в техн. системах"] / В. Давыдов. СПб.: БХВ-Петербург, 2005. - 654 с. ISBN 5-94157-605-6. Экземпляры: всего 17.	17
2.	Давыдов, Владимир Григорьевич. Программирование и основы алгоритмизации [Текст] : [учеб. пособие для студентов вузов по специальности "Упр. и информатика в	10

	техн. системах"] / В. Г. Давыдов. Изд. 2-е, стер. М.: Высшая школа, 2005. - 448 с. ISBN 5-06-004432-7. Экземпляры: всего 10.	
3.	Павловская, Татьяна Александровна. С++. Объектно-ориентированное программирование [Текст] : практикум : учеб. пособие для студентов вузов по напр. подгот. дипломир. спец. "Информатика и вычислит. техника" / Т. А. Павловская, Ю. А. Щупак. СПб.: Питер, 2006. - 264 с. ISBN 5-94723-842-X. Экземпляры: всего 38.	38
4.	Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня для магистров и бакалавров [Текст] : [учебник для студентов вузов по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника"] / Т. А. Павловская. Санкт-Петербург: Питер, 2012. - 460 с. ISBN 978-5-94723-568-5. Экземпляры: всего 13.	13
5.	Тузовский, Анатолий Федорович. Объектно-ориентированное программирование [Текст : Электронный ресурс] : учебное пособие для вузов / А. Ф. Тузовский. Москва: Юрайт, 2022. - 206 с ISBN 978-5-534-00849-4.	<a href="https://urait.ru/bcode/490369">https://urait.ru/bcode/490369</a>

## 6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	519 (III)	Системный блок CEL D-341 FAN/ASUS S-775/512 M/160.0G/DVD+-RW (7), Монитор 15" Samsung 510 M (1), Монитор 17" BenQ FP 71G (1), Монитор TET 20" Samsung SIM 2043W (1), ОСЦИЛЛОГРАФ C1-75 (1), ПК ICL RAY H494.1 сист.блок,клавиат,мышь,монитор View Sonic VA2231 WLED WZ1218) (14), Сист. блок CE 331/256*2/PC 3200/80 Gb/FDD/DVD-ROM/КЛАВ+МЫШЬ+коврик (1),	Microsoft Office Standard, Агент Dr.Web, Комплект ПО для решения основных пользовательских задач

## Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и

полнота воспроизведения учебного материала);  
 - умение применять теоретические знания при решении практических заданий.  
 Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

#### 7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

#### 7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Пороговый уровень

1. Объектно-ориентированное программирование характеризуется:

1. Наличием одной линейной программы
  1. Разделением программы на модули
  2. все данные об объекте, его связи с другими объектами объединяются в одну структурную переменную
2. Метод это:
1. Функция, получая в качестве обязательного параметра указатель на объект
  1. Структура, хранящая указатели this, parent, super
  2. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
  3. Определенный программистом абстрактный тип данных
3. Объект это:
1. Функция, получая в качестве обязательного параметра указатель на объект
  1. Структура, хранящая указатели this, parent, super
  2. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
  3. Определенный программистом абстрактный тип данных
4. Класс это:
1. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
  1. Определенный программистом абстрактный тип данных
  2. Переменная, описанная абстрактным типом данных
5. Когда данные являются не глобальными, доступными всей программе, а локальными доступными только малой части программы:
1. При инкапсуляции
  1. При полиморфизме
  2. При объявлении метода

### **Продвинутый уровень**

6. Когда функция определяется независимо в каждом производном классе и имеет в этих классах общее имя:
1. При инкапсуляции
  1. При полиморфизме
  2. При наследовании и инкапсуляции
7. Для выполнения операции присвоение одного объекта другому:
1. Достаточно чтобы в классе этих объектов был специальный конструктор - конструктор копирования

2. Классы этих объектов нужно описать одинаковым образом
  3. Достаточно чтобы в классе этих объектов были конструкторы и деструкторы
  4. Необходимо описать оба объекта одним именем класса и иметь в классе конструктор копирования
8. В каких случаях надо иметь в классе конструктор копирования:
1. Когда нужно передать элементы класса
  2. Для выполнения операции присвоение одного элемента другому
  3. Для создания дружественной функции
  4. Для уничтожения объекта из памяти
9. В чем состоит задача конструктора копирования:
1. Четко определить порядок создания копии объекта
  2. Размножать объект в указанном количестве копий
  3. Копировать свойства других объектов
  4. Создать this указатель
10. Какое ключевое слово должно быть использовано чтобы указать что объект не является модифицируемым и любая попытка изменения этого типа является ошибкой:
1. readonly
  2. const
  3. static
  4. undeclare
11. Как называются элементы класса которые относятся ко всем экземплярам объектов класса
1. Статические
  2. Динамические
  3. Константные
  4. Защищенные
12. В каких случаях следует вводить статические элементы массива:
1. Когда требуется контроль количества общих методов класса
  2. Когда требуется доступ к элементам массива из объектов дружественного класса
  3. Когда требуется одновременный доступ ко всем объектам класса
  4. Когда требуется определить изменяемые элементы класса
13. Назовите преимущества классов!
1. удобный способ моделирования объектов реального мира
  2. все указанные ответы

3. инкапсуляция данных
4. удобство повторного использования кода

### **Высокий уровень**

14. Назовите принципы объектно-ориентированного представления программных систем

1. абстрагирование; модульность; иерархическая организация
2. абстрагирование; инкапсуляция; модульность; иерархическая организация
3. абстрагирование; инкапсуляция; модульность;

15. Может ли деструктор иметь аргументы?

1. нет
2. да
3. один аргумент типа ссылка на класс

16. Что такое деструктор?

1. Деструктор - это специальная функция-элемент, которая должна уничтожать экземпляр класса после завершения его работы
2. Деструктор - это функция, которая должна открывать динамическую область для экземпляра класса
3. Деструктор - это специальная функция-элемент, которая должна отслеживать данные в экземпляре класса в процессе работы

17. Какие функции есть у любого класса?

1. конструктор и деструктор
2. деструктор
3. нет таких
4. конструктор

18. Понятие this в классе

1. Объект this является аргументом метода другого класса, существует во всех методах и указывает на адрес: this -><объект>
2. Указатель this является скрытым аргументом метода, существует во всех методах объекта и указывает на его (объект) адрес: this ->< объект >
3. Указатель this является скрытым аргументом метода, превращает функцию в область памяти только для чтения.

19. Какого спецификатора доступа в классах нет?

1. public
2. hidden
3. protected

4. private

20. Какой модификатор видимости применяется при наследовании свойств базового класса по умолчанию?

A. protected.

B. public.

C. private.

D. не применяется.

Перечень вопросов для проведения промежуточной аттестации

### ВОПРОСЫ ДЛЯ ЭКЗАМЕНА

1. Понятия объекта и класса. Инкапсуляция.

3. Основные этапы проектирования и разработки программной системы.

4. Классы. Методы класса (функции-члены).

5. Технологические средства разработки ПО. Интегрированные среды.

6 ООП. Дружественность.

7. Средства отладки в интегрированных средах.

8. Уровни инкапсуляции.

9 Диалоговые отладчики.

10. Конструкторы и деструкторы.

11. Средства разработки сложных программных систем.

12. Полиморфизм. Перегрузка функций.

13. Средства и способы поддержки проектов в современных интегрированных средах.

14. Полиморфизм. Перегрузка операций.

15. Генераторы приложений.

16. Наследование. Базовые классы и классы-потомки.

17. Тестирование и оценка качества программных продуктов.

18. Уровни инкапсуляции при наследовании.

19. Виртуальные функции.

20. Этапы конструирования программы.

21 Абстрактные классы.

22. Нисходящее и восходящее проектирование.

23. Конструкторы базовых классов.

24. Процедурная и функциональная декомпозиции при проектировании.

25. Библиотеки классов. Иерархия классов.
26. Объектная декомпозиция при проектировании.
27. Множественное наследование.
28. Основные принципы объектно-ориентированного подхода(ООП).
29. Инкапсуляция. Понятие объекта.
30. Технологические средства разработки ПО. Интегрированные среды.

#### ВОПРОСЫ ДЛЯ ЗАЧЕТА

5. Парадигма ООП.
6. Событийно-ориентированное программирование.
7. Понятие объекта
8. Принцип Инкапсуляции.
9. Определение класса.
10. Данные класса
11. Методы класса (функции-члены).
12. Уровни инкапсуляции.
13. Конструктор класса.
14. Деструктор класса.
15. Принцип Полиморфизма.
16. Перегрузка функций.
17. Множественные конструкторы класса.
18. Конструктор копирования.
19. Перегрузка операций.
20. Шаблоны. Шаблонные функции.
21. Шаблонные классы.
22. Дружественные функции.
23. Перегрузка ввода вывода.
24. Виртуальные функции.
25. Абстрактные классы.
26. Принцип Наследования.
27. Базовые классы и производные классы
28. Уровни инкапсуляции при наследовании.
29. Конструктор производного класса.
30. Иерархии классов.



